CHALMERS
UNIVERSITY OF TECHNOLOGY

Multiple Objective Functions for Falsification of Cyber-Physical Systems

Multiple Objective Functions for Falsification of Cyber-Physical Systems

Zahra Ramezani * **Johan Lidén Eddeland** * **Koen Claessen** **
Martin Fabian * **Knut Åkesson** *

* *Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden (e-mails: {rzahra, johan.eddeland, fabian, knut}@chalmers.se).*
** *Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Sweden (e-mail: koen@chalmers.se)*

Abstract:
Cyber-physical systems are typically safety-critical, thus it is crucial to guarantee that they conform to given *specifications*, that are the properties that the system must fulfill. Optimization-based falsification is a model-based testing method to find counterexamples of the specifications. The main idea is to measure how far away a specification is from being broken, and to use an optimization procedure to guide the testing towards falsification. The efficiency of the falsification is affected by the objective function used to evaluate the test results; different objective functions are differently efficient for different types of problems. However, the efficiency of various objective functions is not easily determined beforehand. This paper evaluates the efficiency of using multiple objective functions in the falsification process. The hypothesis is that this will, in general, be more efficient, meaning that it falsifies a system in fewer iterations, than just applying a single objective function to a specific problem. Two objective functions are evaluated, *Max*, *Additive*, on a set of benchmark problems. The evaluation shows that using multiple objective functions can reduce the number of iterations necessary to falsify a property.

Keywords: Testing, Falsification, Multiple Objective Functions, Cyber-Physical Systems.

1. INTRODUCTION

Cyber-Physical Systems (CPSs) consist of computational parts described by state models, communicating with a physical environment described by differential equations. Using high fidelity models is typical in *model-based design* of CPSs. Testing and verifying the correctness of all physical and cyber components of CPSs are important and a big challenge (Abbaspour Asadollah et al., 2015). An autonomous car is an example of a CPS where it is necessary with rigorous methods to assure the correctness of the system. Typically, *formal verification* and/or *testing* are used for this purpose.

For complex systems, testing is a necessary part of the design process since formal verification of systems with a combination of discrete and continuous dynamics is an undecidable problem (Henzinger et al., 1995). However, for both formal verification and testing, formal specifications of the properties that should be fulfilled are required in order to enable an automated approach. One approach, surveyed by Bartocci et al. (2018), that can be used is formal specification of properties that the closed-loop system should satisfy, combined with simulation of models and where the given specifications are monitored and it is evaluated whether they are satisfied or not. This can be combined with *falsification* techniques that search for counterexamples to given specifications of the closed-loop system.

The falsification process can be based on an optimization procedure where the optimization is performed over an input parametrization expressing possible input signals. The aim is to find counterexamples, if possible, to specifications of the system under test. This is done in an iterative manner where an objective function measures the distance to the specification being falsified. The objective function is determined by the definition of quantitative semantics for temporal logic formalisms (Fainekos and Pappas, 2009). Metric Interval Temporal Logic (MITL) (Koymans, 1990) and Signal Temporal Logic (STL) (Maler and Nickovic, 2004), are two variants of formalisms for which quantitative semantics can be defined. The main purpose of calculating an objective function value is to guide the testing process towards falsification by choosing the next set of parameters for the input signals to the system being simulated, such that the likelihood of falsifying the specification is increased.

For industrial systems typically only a *black-box* of the system under test is available, meaning that only input-output behavior of the system can be observed. In general, these systems are a mix of continuous dynamics, discrete event dynamics, and algorithms implemented using general-purpose programming languages. Parts of the system might also be implemented using physical hardware. Breach (Donzé, 2010) and S-TaLiRo (Annpureddy et al., 2011) are two Matlab/Simulink based toolboxes used for test monitoring and falsification. Both tools search for trajectories of minimal quantitative value to find counterexamples to MITL/STL specifications. Eddeland et al. (2020) show how STL specifications can be automatically de-

⋆ This work was supported by the Swedish Research Council (VR) project SyTeC VR 2016-06204 and from the Swedish Governmental Agency for Innovation Systems (VINNOVA) under project TESTRON 2015-04893.

rived from Simulink blocks expressing the specifications, this is of practical value to engineers that are working with testing and falsification since it is not necessary to work with temporal logic specifications directly.

For both Breach and S-TaLiRo, the falsification process is guided by an optimization procedure. Due the system being a black-box, gradient-free optimization methods have to be used. Nelder-Mead (Nelder and Mead, 1965) is a common gradient-free optimization method that can be used by Breach to guide the falsification process. Although the optimization method does not use explicit gradients, the method will attempt to search in a direction that results in a smaller objective value, where the quantitative semantics are defined in such a way that a negative objective value means that the specification is falsified. In this work, we evaluate three quantitative semantics, *Max* and *Additive* to define the objective function for the falsification processes. In previous work, (Claessen et al., 2018), Valued Booleans (VBools) were introduced as a way to express different quantitative semantics in a coherent way.

In Ramezani et al. (2019), the *Max* and MARV semantics for defining objective functions were evaluated for an autonomous driving example. It was shown that, for certain areas of the parameter space, *Max* results in constant objective values, while MARV results in non-constant objective values. If the Nelder-Mead (NM) solver starts in an area where the objective values are constant (like for *Max*), it might eventually finish the optimization procedure without finding any falsifying point. This happens because there is no useful information for the optimization algorithm to guide the search to areas with parameters where the objective function has a lower value. As the simulation time is the most limiting factor; the more simulations that have to be run, the longer the falsification process takes.

The contribution of this paper is the introduction of a modified optimization approach that takes advantage of multiple objective functions for the purpose of falsifying specifications. The objective functions have in common that if the specification is satisfied the value of the objective function is positive and if the specification is falsified the value is negative. The motivation for this work is that evaluating multiple objective functions is often significantly less time-consuming than simulating or executing the system, and the objective values of multiple quantitative semantics can be computed using a single simulation of the system under test. The modified optimization approach then heuristically chooses which one of the parameter configurations to simulate next based on the variance of the respective objective function values. That is, for each iteration of the NM solver, the heuristic picks the point given by the quantitative semantic that has the largest variance. This avoids using the semantic that has close to constant objective values. The approach is evaluated on a set of benchmark examples. The results show that using multiple objective functions can indeed falsify system properties in fewer simulation runs, compared to using only a single objective function.

In the following, Section 2 introduces the quantitative semantics and different ways to define the objective functions used for the falsification process. Section 3 proposes the suggested multiple objective functions in this paper. Section 4 introduces the three benchmark examples. Section 5 evaluates the performance of the suggested optimization on benchmark examples. Finally, Section 6 summarizes the contributions.

2. QUANTITATIVE SEMANTICS AND OBJECTIVE FUNCTIONS

In this paper, Breach is used for falsification, hence STL is used to model the specifications. The syntax of STL is defined as follows (Raman et al., 2014)

$$\varphi ::= \mu \mid \neg\mu \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \Box_{[a,b]}\psi \mid \Diamond_{[a,b]}\psi$$

where the predicate μ is $\mu \equiv \mu(s) > 0$ and s is a signal; φ and ψ are STL formulas; $\Box_{[a,b]}$ denotes the *globally* operator between times a and b (with $a < b$); $\Diamond_{[a,b]}$ denotes the *finally* operator between a and b.

The satisfaction of the formula φ with respect to the discrete signal s at the discrete time instant k is defined as:

$$\begin{aligned}
(s,k) &\models \mu && \Leftrightarrow\ \mu(s[k]) \\
(s,k) &\models \neg\mu && \Leftrightarrow\ \neg((s,k) \models \mu) \\
(s,k) &\models \varphi \wedge \psi && \Leftrightarrow\ (s,k) \models \varphi \wedge (s,k) \models \psi \\
(s,k) &\models \varphi \vee \psi && \Leftrightarrow\ (s,k) \models \varphi \vee (s,k) \models \psi \\
(s,k) &\models \Box_{[a,b]}\varphi && \Leftrightarrow\ \forall k' \in [k+a, k+b], (s,k') \models \varphi \\
(s,k) &\models \Diamond_{[a,b]}\varphi && \Leftrightarrow\ \exists k' \in [k+a, k+b], (s,k') \models \varphi
\end{aligned}$$

Instead of only checking the boolean satisfaction of an STL formula, the notion of a quantitative value, i.e. an objective value, will be defined in order to measure how far away a specification is from being falsified. A Valued Boolean (VBool) (Claessen et al., 2018) (v, x) is a combination of a Boolean value v (true $\top$, or false $\bot$) together with a real number x that is a measure of how true or false the specification is. This value will be used as a measure of how convincingly a test passed, or how severely it failed, respectively. In the original VBool definition, x is defined to always be non-negative. However, in this paper we use the convention that x is negative when v is false, and positive otherwise.

2.1 Quantitative Semantics

Using VBools, we define three quantitative semantics: *Max*, which is essentially the same as standard STL quantitative semantics; *Additive*; For these semantics we define the respective *and*, *or*, *always*, and *eventually* operators.

For conjunction, the semantics differ only in the two cases where the truth values are the same:

	Max	*Additive*
$(\top, x) \wedge (\top, y) =$	$(\top, \min(x, y))$	$\left(\top, \frac{1}{\frac{1}{x}+\frac{1}{y}}\right)$
$(\top, x) \wedge (\bot, y) =$	$(\bot, y)$	
$(\bot, x) \wedge (\top, y) =$	$(\bot, x)$	
$(\bot, x) \wedge (\bot, y) =$	$(\bot, \max(x, y))$	$(\bot, x + y)$

Using the de Morgan laws, the *or* operator can be defined in terms of *and*, as $(v_x, x) \vee (v_y, y) = \neg_v(\neg_v(v_x, x) \wedge \neg_v(v_y, y))$, where VBool negation is defined as $\neg_v(v_x, x) = (\neg v_x, -x)$.

For the *Max* semantics, the *always* operator over an interval $[a, b]$ is straightforwardly defined in terms of *and*, as $\Box_{[a,b]}\varphi = \bigwedge_{k=a}^{b} \varphi[k]$, where φ is a finite sequence of VBools defined for all the discrete time instants in $[a, b]$.

For the *Additive* semantics, though, *always* is a bit more elaborate: $\Box_{[a,b]}\varphi = \bigwedge_{k=a}^{b} \varphi[k] \# \delta t$, where δt is the simulation step size

that makes the quantitative value independent of the simulation time, and # is $(\bot, x) \# \delta t = (\bot, x \cdot \delta t)$ and $(\top, x) \# \delta t = (\top, x/\delta t)$. Furthermore, the *eventually* operator is for all three semantics defined over an interval $[a, b]$ in terms of *always*, as $\Diamond_{[a,b]}\varphi = \neg(\Box_{[a,b]}(\neg_v \varphi))$.

3. FALSIFICATION USING MULTIPLE OBJECTIVE FUNCTIONS

This section presents the multiple objective functions for falsification of CPSs. Different combinations of objective functions and different strategies for switching between objective functions is discussed in this paper. The main optimization algorithm considered in this paper is an implementation of Nelder-Mead which is also included in Breach. This algorithm is implemented as *fminsearch* (Lagarias et al., 1998) in Matlab. We propose a modified optimization algorithm, based on Nelder-Mead, that exploits multiple objective functions. The new algorithm is presented in Algorithm 1 and works in the following way. The presentation of the algorithms is based on two objective functions, in this case using the *Max* and *Additive* semantics, however the approach is generic and can be applied to an arbitrary number of objective functions.

(1) The algorithms starts with p sample points. For each example, p is different and it refers to the number of inputs of each example multiple by 10. All these points are sorted from lowest to highest according to the values of different objective functions. Note, that objective values will be different for each used quantitative semantic, thus there will be one ordering for each semantic. By using a heuristic algorithm to choose which quantitative semantic, the minimum point is considered and it will generate n new points surround the first point before starting the optimization.
(2) The first $n+1$ points in the parameter space are needed by the NM style optimization algorithm to start the optimization process. Again, all these points must be sorted from lowest to highest according to the values of the different objective functions.
(3) For each ordering of the objective values, the algorithm will suggest a new point in the parameter space for evaluation. For each quantitative semantic there will be an ordering of the points in the parameter space that is based on the objective value used for the specific semantic. By using a heuristic algorithm to choose which quantitative semantic, one new point will be selected for further evaluation, i.e. being simulated.
(4) Now, one iteration in the optimization can execute, i.e. the execution of Step 4 to 7 in Algorithm 1.
(5) When reaching to Step 8, once again calculate the objective values for *all* of the considered quantitative semantics and saved points, and create one ordering for each of the considered quantitative semantic. Choose $n+1$ points with lowest objective value function of the semantic that wins the heuristic strategy.

Note, a new quantitative semantic can be selected in each iteration of the algorithm. In this paper, we have implemented two heuristic algorithms that are based on the variance of the $n+1$ lowest objective values and the distance of largest and lowest objective functions.

Strategy 1 (Variance). For each ordering of objective values, i.e. one ordering for each quantitative semantic, consider the $n+1$ points with lowest objective value, i.e. the points that according to the semantics that are closest to falsifying the specification. For these points, calculate the variance among the $n+1$ points using $\sigma_k^2 = \sum_{i=1}^{n+1} \frac{(f_i^k - \mu)^2}{n+1}$, where f_i^k refers to the objective function value of each $n+1$ points that have lowest value for each semantic. μ is the mean of $n+1$ points.

Strategy 2 (Distance). The distance can be calculated using $Dis_k = \frac{f_{max}^k - f_{min}^k}{\|x_{max}^k - x_{min}^k\|}$, where f_{max}^k and f_{min}^k refer to the maximum and minimum objective function values of the $n+1$ points that have the lowest objective value of each semantic. x_{max}^k and x_{min}^k refers to their points, respectively.

The heuristic will choose the point given by the quantitative semantic that corresponds to the largest variance in strategy 1; largest distance in strategy 2. These heuristics are thus selecting the quantitative semantic that has a clear sense of direction and avoids using semantic that has close to constant objective values resulting in small variance and distance. However, other heuristics could be used as well, but a more thorough evaluation of possible heuristics is future research.

4. BENCHMARK PROBLEMS

Three examples are considered here to show the performance of the multiple objective functions approach, that are also used in Eddeland et al. (2020), below is a brief description of the examples.

4.1 Automatic Transmission (AT) Benchmark

The inputs to the model are the throttle and brake of a vehicle. The outputs of the model are the vehicle speed v, the engine speed ω, and the gear, see Hoxha et al. (2014) for details.

4.2 Third Order $\Delta - \Sigma$ Modulator

The third order $\Delta - \Sigma$ modulator is a model of a technique for analog to digital conversion. It has one input U, three states x_1, x_2, x_3, and three initial conditions $x_1^{init}, x_2^{init}, x_3^{init}$, see Dang et al. (2004) for details.

4.3 Static Switched (SS) System

The static switched system is a model without any dynamics that is included as a simple case make falsification worse than only using single boolean objective function. The model has been inspired by Dokhanchi et al. (2015).

5. EXPERIMENTAL SETUP AND RESULTS

The experimental setup is described in more detail in Eddeland et al. (2020). The implementation starts by evaluating 100 random points for the AT example, 40 for third order $\Delta - \Sigma$ modulator example and 20 for the SS example before starting the optimization. After doing that if the falsified point is not found, the optimization solver starts from the point with lowest objective value. In Table 1, the STL specifications that should be falsified for all the models, and the benchmark models are presented.

The results of running Algorithm 1 on the benchmark problems are shown in Tables 2, 3, and 4, and the aggregated results are

Algorithm 1 Modified Nelder-Mead Algorithm Using Multiple Objective Functions

1. Choose p random points. Start with p random points and order and re-label the vertices from lowest function value to highest function value:
$f^{Max}(x_1^1) \leq f^{Max}(x_2^1) \leq \cdots \leq f^{Max}(x_p^1)$,
$f^{Add}(x_1^2) \leq f^{Add}(x_2^2) \leq \cdots \leq f^{Add}(x_p^2)$,
Use a heuristic algorithm to choose which quantitative semantic to follow in this iteration of the optimization. Take the minimum point of semantic that wins the heuristic algorithm.
2. Let x_i denote the list of vertices in the current simplex, $i = 1, \ldots, n+1$. These points are generated from the minimum point of Step 1.
3. Order. For each objective function $i = 1, \ldots, j$, order and re-label the $n+1$ vertices from lowest function value to highest function value:
$f^{Max}(x_1^1) \leq f^{Max}(x_2^1) \leq \cdots \leq f^{Max}(x_{n+1}^1)$,
$f^{Add}(x_1^2) \leq f^{Add}(x_2^2) \leq \cdots \leq f^{Add}(x_{n+1}^2)$,
Use a heuristic algorithm to choose which quantitative semantic to follow in this iteration of the optimization, see Strategy 1 and 2 for examples.
4. Reflection. Compute the reflected point x_r by $x_r = \bar{x} + \rho(\bar{x} - x_{(n+1)})$, where $\bar{x}$ is the centroid of the n points with lowest objective function values, $\bar{x} = \sum \frac{x_i}{n}$, $i = 1, \ldots, n$. The rest of the optimization will be executed with the chosen semantic, then f refers to the objective function value of that semantic.
if $f(x_1) < f(x_r) < f(x_n)$ **then**
  Replace x_{n+1} with the point x_r and go to Step 8.
end if
5. Expansion.
if $f(x_r) < f(x_1)$ **then**
  Compute the expanded point x_e by $x_e = \bar{x} + \chi(x_r - \bar{x})$.
  if $f(x_e) < f(x_1)$ **then**
    Replace x_{n+1} with x_e and go to Step 8.
  else
    Replace x_{n+1} with x_r and go to Step 8.
  end if
end if
6. Contraction.
if $f(x_r) \geq f(x_n)$ **then**
  Perform a contraction between $\bar{x}$ and the best among x_{n+1} and x_r.
  if $f(x_n) \leq f(x_r) < f(x_{n+1})$ **then**
    Calculate $x_{oc} = \bar{x} + \tau(x_r - \bar{x})$ *Outside contract*.
    if $f(x_{oc}) \leq f(x_r)$ **then**
      Replace x_{n+1} with x_{oc} and go to Step 8.
    else
      Go to Step 7.
    end if
  end if
end if
if $f(x_r) \geq f(x_{n+1})$ **then**
  Calculate $x_{ic} = \bar{x} + \tau(x_{n+1} - \bar{x})$ *Inside contract*.
  if $f(x_{ic}) \geq f(x_{(n+1)})$ **then**
    Replace x_{n+1} with x_{ic} and go to Step 8.
  end if
end if
7. Shrink. Evaluate the n new vertices $x' = x_1 + \phi(x_i - x_1)$, $i = 2, \ldots, n+1$. Replace the vertices $x_2, ..., x_{n+1}$ with the new vertices $x'_2, ..., x'_{n+1}$.
8. Re-Order. Calculate the objective values for the new point for *all* the quantitative semantics and for each quantitative semantic and save it or them (If "Shrink" happens). Order and re-label the vertices of all m calculated points from lowest function value to highest function:
$f^{Max}(x_1^1) \leq f^{Max}(x_2^1) \leq \cdots \leq f^{Max}(x_m^1)$,
$f^{Add}(x_1^2) \leq f^{Add}(x_2^2) \leq \cdots \leq f^{Add}(x_m^2)$,
where k refers to the number of $n+1$ of Step 2 and plus the number of points that are reached at each the iterations (Steps 4-7).
9. Selecting semantics. Take $n+1$ of each semantic that has lowest objective function values from Step 8. Select the semantic according to the heuristic algorithm and continue with the chosen semantic, f refers to the objective function value of the chosen semantic. While the stopping condition is not reached go to Step 4, thus
if $f(x_{n+1}) - f(x_1) < \epsilon$ **then**
  Stop, where $\epsilon > 0$ is a small predetermined tolerance.
else
  Go to Step 4.
end if
Note: ρ, χ, τ are constant parameters.

Table 1. Specifications to falsify for the three benchmark models AT, $(\Delta - \Sigma)$, and SS.

Spec.	Formula
φ_1^{AT}	$\Diamond_{[0,T]}(\omega \geq 2000)$
φ_2^{AT}	$\Box\Diamond_{[0,T]}(\omega \leq 3500 \vee \omega \geq 4500)$
φ_3^{AT}	$\Box_{[0,T]}(\neg(gear == 4))$
φ_4^{AT}	$\Diamond(\Box_{[0,T]}(gear == 3))$
φ_5^{AT}	$\bigwedge_{i=1,...,4} \Box((\neg(gear == i) \wedge \Diamond_{[0,\epsilon]}(gear == i) \implies (\Box_{[\epsilon,T+\epsilon]}(gear == i)))$
φ_6^{AT}	$\Box_{[0,T]}(v \leq 85) \vee \Diamond(\omega \geq 4500)$
φ_7^{AT}	$\neg\Big((\Box_{[0,1]}gear == 1) \wedge (\Box_{[2,4]}gear == 2) \wedge (\Box_{[5,7]}gear == 3) \wedge (\Box_{[8,10]}gear == 3) \wedge (\Box_{[12,15]}gear == 2)\Big)$
φ_8^{AT}	$\Box_{[0,20]}((gear == 4 \wedge throttle > 45 \wedge throttle < 50) \implies \omega < \bar{\omega})$
$\varphi^{\Delta-\Sigma}$	$\Box\Big(\bigwedge_{i=1}^{3}(-1 \leq x_i \wedge x_i \leq 1)\Big)$.
φ^{SS}	$\Box(y \geq 0)$

shown in Fig. 1. The tables are formatted as follows. The first column denotes the specification falsified. Each specification has one to three parameter values, these parameter values are shown in the second columns. The remaining columns show the different semantics including the *Max*, *Additive*, and their combination with two difference strategies. For each parameter value and semantic, two values are presented. The first value is the relative success rate of falsification, in percent. There are a total of 20 falsification run for each parameter value and objective functions, meaning that the success rate will be a multiple of 5%. The second value, inside parentheses, is the average number of needed simulations *per successful falsification*. Each falsification is set to have a maximum of 1000 simulations performed.

In addition, for each parameter value, the semantic with the highest (or tied highest) success rate has the success rate displayed in **bold** characters. For each parameter value if there are semantics with same success rate, the semantic with the lowest average number of simulations per successful simulation has that number displayed in **bold** characters (inside the parentheses).

5.1 Results

As can be seen from Fig 1, the top two approaches are Multiple semantics. All two strategies of the multiple objective functions, perform better than when we only have a single objective function. They are more successful in falsifying with fewer simulations.

By looking at three tables of results, it can been seen that for automatic transmission, φ_1^{AT} $(T = 20)$, φ_3^{AT} $(T = 5)$, φ_5^{AT} $(T = 2)$, φ_6^{AT} $(T = 12)$, φ_8^{AT} $(\bar{\omega} = 3000)$, the Multi-Max-Add using variance strategy works better. While, for φ_1^{AT} $(T = 30)$, φ_2^{AT}, φ_4^{AT}, φ_5^{AT} $(T = 1)$, φ_6^{AT} $(T = 12)$, φ_7^{AT}, the Multi-Max-Add using distance works better. Only, for φ_3^{AT} $(T = 4.5)$, *Max* is better. *Additive* works for φ_1^{AT} $(T = 40)$, φ_8^{AT} $(\bar{\omega} = 3500)$.

For $\varphi^{\Delta-\Sigma}$ Benchmark, except $U \in [-0.45, 0.45]$ that *Max* performs better for that, multiple objective function using variance perform well. For all specifications of the Static Switched

Table 2. Results for the automatic transmission benchmark. For each parameter value and quantitative semantics, the first number indicates relative success ratio of falsification (%). The second number, in parentheses, indicates average number of simulations per successful falsification.

Spec.	Parameters \ Semantics	Max	Add	Mul Max-Add (Variance)	Mul Max-Add (Distance)
φ_1^{AT}	$T = 20$	**100** (138)	**100** (156)	**100** (93)	**100** (134)
	$T = 30$	85 (264)	95 (364)	95 (375)	**100** (309)
	$T = 40$	35 (315)	65 (556)	55 (561)	50 (483)
φ_2^{AT}	$T = 10$	**100** (33)	**100** (14)	**100** (20)	**100** (11)
φ_3^{AT}	$T = 4.5$	**100** (141)	90 (323)	**100** (223)	**100** (272)
	$T = 5$	**100** (65)	**100** (95)	**100** (44)	**100** (67)
φ_4^{AT}	$T = 1$	60 (505)	35 (357)	40 (367)	65 (402)
	$T = 2$	**100** (21)	**100** (19)	**100** (19)	**100** (14)
φ_5^{AT}	$T = 1$	95 (406) 90	75 (516)	95 (333)	**100** (330)
	$T = 2$	**100** (5)	**100** (4)	**100** (4)	**100** (6)
φ_6^{AT}	$T = 10$	50 (722)	45 (482)	55 (534)	55 (526)
	$T = 12$	**100** (236)	**100** (215)	**100** (186)	**100** (198)
φ_7^{AT}		65 (766)	75 (382)	65 (483)	90 (421)
φ_8^{AT}	$\bar{\omega} = 3000$	**100** (13)	**100** (11)	**100** (7)	**100** (10)
	$\bar{\omega} = 3500$	30 (439)	90 (375)	60 (372)	15 (323)

Table 3. Results for the Third Order $\Delta - \Sigma$ modulator. For each parameter value and quantitative semantics, the first number indicates relative success ratio of falsification (%). The second number, in parentheses, indicates average number of simulations per successful falsification.

Spec.	Parameters \ Semantics	Max	Add	Mul Max-Add (Variance)	Mul Max-Add (Distance)
$\varphi^{\Delta-\Sigma}$	$U \in [-0.35, 0.35]$	55 (331)	20 (515)	85 (361)	65 (445)
	$U \in [-0.40, 0.40]$	**100** (271)	75 (279)	**100** (228)	**100** (255)
	$U \in [-0.45, 0.45]$	**100** (73)	95 (314)	**100** (136)	**100** (141)

Table 4. Results for the Static Switched System. For each parameter value and quantitative semantics, the first number indicates relative success ratio of falsification (%). The second number, in parentheses, indicates average number of simulations per successful falsification.

Spec.	Parameters \ Semantics	Max	Add	Mul Max-Add (Variance)	Mul Max-Add (Distance)
φ^{SS}	$thresh = 0.7$	**100** (120)	**100** (248)	**100** (43)	**100** (89)
	$thresh = 0.8$	90 (201)	**100** (219)	**100** (114)	85 (356)
	$thresh = 0.9$	55 (511)	45 (519)	80 (334)	40 (365)

system, multiple objective function using variance give better results.

One conclusion that can be given here is that the Max-Additive multiple objective function using both strategies work better than others, and for only a few of specifications, the single semantics works better. Only for the specifications φ_8^{AT} ($\bar{\omega} = 3500$) and φ_5^{AT} ($T = 30$) of the automatic transmission example, the *Additive* was successful in finding more falsification. For other semantics that *Max* or *Additive* work better still the single and multiple objective functions have same success ratio of falsification, only the number of simulations is different. As a result, the multiple objective functions using both strategies can guide the testing process towards falsification better than only single semantic is used, on the given benchmark set.

6. CONCLUSION

In this paper, the use of multiple objective functions for falsification of CPSs was proposed. With a single objective function, the optimization may get the wrong or even no information for it to be able to guide the falsification process towards falsifying the specification. Since for CPSs, the simulation time is the most limiting factor, not the evaluation of the simulation results, multiple objective functions were suggested. Combinations of two different semantics *Max* and *Additive* were evaluated. A variance and distance based strategy were used to switch between the objective functions, such that the objective function with highest variance (distance) was picked for each iteration of the falsification. Three benchmark CPSs examples were considered to show the performance of the multiple objective functions.

The main conclusion drawn from the data gathered is that the proposed optimization algorithm perform better than when only a single objective function is used on the used benchmark examples. Also multiple objective functions are more successful in finding counterexamples in less number of simulation runs. This so, since multiple objective functions can better guide the optimization algorithm towards falsification, increasing the chance of falsifying the specification.

For future work, it would be interesting to explore different heuristics for choosing between multiple objective functions as well as extending the number of benchmark problems to further evaluate the performance of the proposed approach.

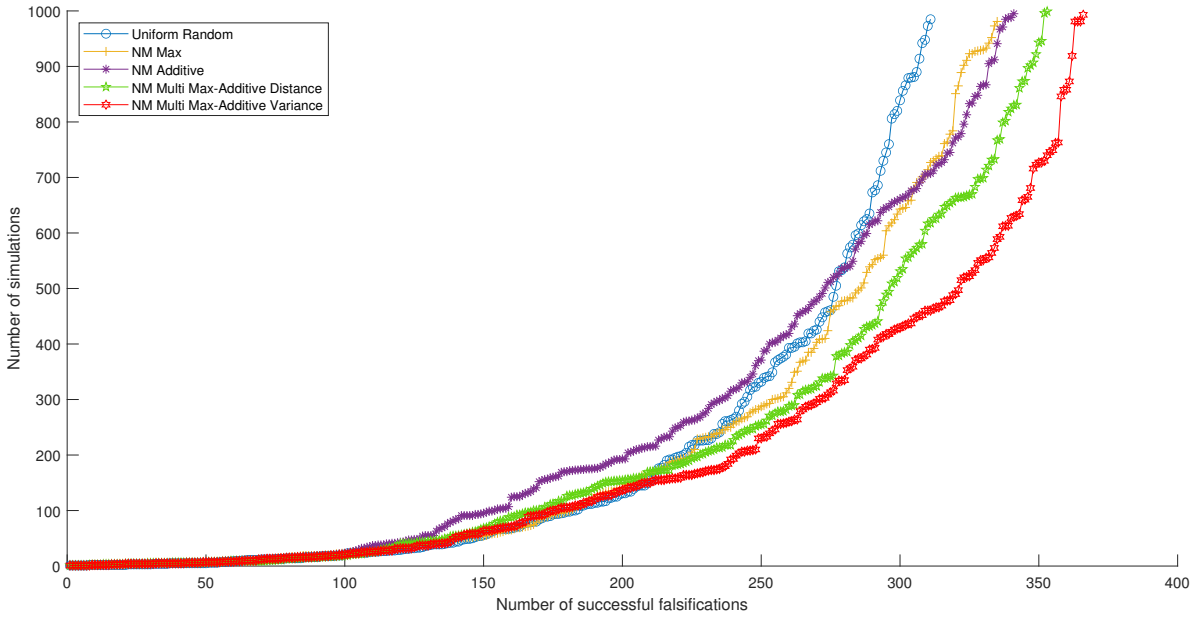

Fig. 1. A cactus plot showing performance of using optimization based on multiple objective functions compared to using a single objective function. The plotted values tell how many successful falsifications (x-axis) were completed in less than a specific number of simulations (y-axis). The maximum number of simulations per falsification is 1000. Note that we also include the cactus plot for Uniform Random sampling, as a baseline approach.

REFERENCES

Abbaspour Asadollah, S., Inam, R., and Hansson, H. (2015). A survey on testing for cyber physical system. In K. El-Fakih, G. Barlas, and N. Yevtushenko (eds.), *Testing Software and Systems*, 194–207. Springer International Publishing, Cham.

Annpureddy, Y., Liu, C., Fainekos, G., and Sankaranarayanan, S. (2011). S-TaLiRo: A tool for temporal logic falsification for hybrid systems. In P.A. Abdulla and K.R.M. Leino (eds.), *Tools and Algorithms for the Construction and Analysis of Systems*, 254–257. Springer, Berlin, Heidelberg.

Bartocci, E., Deshmukh, J., Donzé, A., Fainekos, G., Maler, O., Nickovic, D., and Sankaranarayanan, S. (2018). Specification-based monitoring of cyber-physical systems: A survey on theory, tools and applications. In *Lectures on Runtime Verification*, volume 10457 of *Lecture Notes in Computer Science*, 135–175.

Claessen, K., Smallbone, N., Eddeland, J., Ramezani, Z., and Åkesson, K. (2018). Using valued booleans to find simpler counterexamples in random testing of cyber-physical systems. *IFAC-PapersOnLine*, 51(7), 408 – 415. 14th IFAC Workshop on Discrete Event Systems WODES 2018.

Dang, T., Donzé, A., and Maler, O. (2004). Verification of analog and mixed-signal circuits using hybrid system techniques. In *International Conference on Formal Methods in Computer-Aided Design*, 21–36. Springer.

Dokhanchi, A., Zutshi, A., Sriniva, R.T., Sankaranarayanan, S., and Fainekos, G. (2015). Requirements driven falsification with coverage metrics. In *Proceedings of the 12th International Conference on Embedded Software*, 31–40.

Donzé, A. (2010). Breach, a toolbox for verification and parameter synthesis of hybrid systems. In T. Touili, B. Cook, and P. Jackson (eds.), *Computer Aided Verification*, 167–170. Springer, Berlin, Heidelberg.

Eddeland, J.L., Claessen, K., Smallbone, N., Ramezani, Z., Miremadi, S., and Åkesson, K. (2020). Enhancing temporal logic falsification with specification transformation and valued booleans. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and System*. Early access.

Fainekos, G.E. and Pappas, G.J. (2009). Robustness of temporal logic specifications for continuous-time signals. volume 410, 4262 – 4291.

Henzinger, T.A., Kopke, P.W., Puri, A., and Varaiya, P. (1995). What's decidable about hybrid automata? In *Proceedings of the Twenty-seventh Annual ACM Symposium on Theory of Computing*, STOC '95, 373–382. ACM, New York, USA.

Hoxha, B., Abbas, H., and Fainekos, G. (2014). Benchmarks for temporal logic requirements for automotive systems. *Proc. of Applied Verification for Continuous and Hybrid Systems*.

Koymans, R. (1990). Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4), 255–299.

Lagarias, J., Reeds, J., Wright, M., and Wright, P. (1998). Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM J. on Optimization*, 9, 112–147.

Maler, O. and Nickovic, D. (2004). Monitoring temporal properties of continuous signals. In Y. Lakhnech and S. Yovine (eds.), *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, 152–166. Springer, Berlin.

Nelder, J.A. and Mead, R. (1965). A Simplex Method for Function Minimization. *The Computer J.*, 7(4), 308–313.

Raman, V., Donzé, A., Maasoumy, M., Murray, R.M., Sangiovanni-Vincentelli, A., and Seshia, S.A. (2014). Model predictive control with signal temporal logic specifications. In *53rd IEEE Conference on Decision and Control*, 81–87.

Ramezani, Z., Smallbone, N., Fabian, M., and Åkesson, K. (2019). Evaluating two semantics for falsification using an autonomous driving example. In *2019 IEEE 17th International Conf. on Industrial Informatics*, volume 1, 386–391.